



Oracle Zero Downtime Migration



ZDM Physical Migration Step by Step Guide

Month August, 2021 | Version 1.02
Copyright © 2021, Oracle and/or its affiliates
PUBLIC

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

TABLE OF CONTENTS

Disclaimer	1
Introduction	3
Zero Downtime Migration	4
Architecture	4
Supported Platforms and configurations	4
Requirements	5
Zero Downtime Migration Service Host Requirements	5
Source Database Requirements	5
Target Database Requirements	6
Backup requirements	7
Oracle NET Services (SQL*Net) Connectivity Requirements	7
Port Requirements	7
ZDM Installation & Pre-Migration Configuration	8
Oracle ZDM Service Node Configuration	8
Configuring Connectivity between ZDM Service Node and Source and Target Database Servers	13
Configuring Connectivity between the Source and Target Database Servers	14
Option 1 - SCAN Connectivity	14
Option 2 - SSH Tunnel	15
Creating a Target Database	15
Setting up a Transparent Data Encryption Wallet	18
Creating Object Storage for Database Backup	19
Generating an Authentication Token	21
Preparing the Response File	23
Response File Parameters	23
Database Migration with ZDM	27
Performing a Database Migration pre-check process	27
Performing a Database Migration	29
Frequently Asked Questions & Common Issues	31
F.A.Q	31
Known Issues	34
Troubleshooting & Other Resources	34

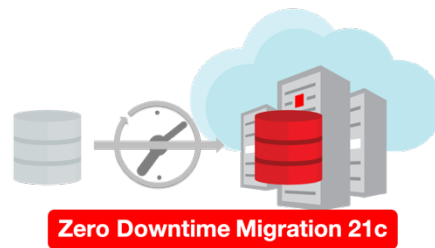


Figure 0 – Oracle Zero Downtime Migration 21c Logo

INTRODUCTION

Oracle customers are moving Oracle workloads into the Oracle Cloud or onto Engineered Systems at a growingly rapid pace. However, migrating workloads has been a source of challenges for many years. In particular, migrating database workloads from one system to another or into the Cloud is easier said than done.

Based on years of experience migrating Oracle workloads, Oracle has developed Zero Downtime Migration (ZDM). ZDM is Oracle's premier solution for a simplified and automated migration experience, providing zero to negligible downtime for the production system and depending on the migration scenario. ZDM allows you to directly and seamlessly migrate your on-premises Oracle Databases to and between any Oracle-owned infrastructure, including Exadata Database Machine On-Premises, Exadata Cloud at Customer (ExaC@C), and Oracle Cloud Infrastructure. Oracle ZDM supports a wide range of Oracle Database versions and, as the name implies, ensures minimal to no production database impact during the migration.

ZDM follows Oracle Maximum Availability Architecture (MAA) principles¹ and incorporates products such as GoldenGate and Data Guard to ensure High Availability and an online migration workflow that leverages technologies such as the Recovery Manager, Data Pump, and Database Links.

This technical brief is a step-by-step guide for migrating your on-premises Oracle Databases to the Oracle Cloud with Zero Downtime Migration's Physical workflow. The scenario used for this migration comprises a source database running on Oracle Cloud Infrastructure DBCS Virtual Machines (emulating an on-premises environment) and a target database running on Oracle Cloud Infrastructure DBCS Virtual Machines.

Oracle ZDM will run on a separate node and connect to both source and target to perform the migration. This guide will cover all requirements related to installing the Oracle ZDM service host, the source database to be migrated, the target database recipient of the migration process, the backup and networking used. The migration process will be dissected and done in a step-by-step fashion. This guide will answer the most frequently asked questions regarding the product and the overall migration process.

For more information on Oracle Zero Downtime Migration, please visit ZDM's product website.²

¹ <http://oracle.com/goto/maa>

² [Http://www.oracle.com/goto/zdm](http://www.oracle.com/goto/zdm)

ZERO DOWNTIME MIGRATION

Architecture

Oracle Zero Downtime Migration (ZDM) is the Oracle Maximum Availability Architecture (MAA)-recommended solution to migrate Oracle Databases to the Oracle Cloud. ZDM's inherent design keeps in mind the migration process as straightforward as possible and to ensure the most negligible impact on production workloads. The source database to be migrated can be on-premises, deployed on Oracle Public Cloud Gen 1 or Oracle Cloud Infrastructure. The target database deployment can be in a Database Cloud Service on Oracle Cloud Infrastructure (OCI) Virtual Machine, Exadata Cloud Service, Exadata Cloud at Customer, or Autonomous Database. ZDM automates the entire migration process, reducing the chance of human errors. ZDM leverages Oracle Database-integrated high availability (HA) technologies such as Oracle Data Guard and GoldenGate and follows all MAA best practices that ensure no significant downtime of production environments. Oracle ZDM supports both Physical and Logical Migration workflows. This technical brief covers a step-by-step guide for the Logical Migration Workflow leveraging the Object Storage as a backup location.

ZDM Physical Migration Workflow performs a zero downtime migration in eight simple steps, which can be scheduled and monitored as needed. The following section provides details on each of these steps.

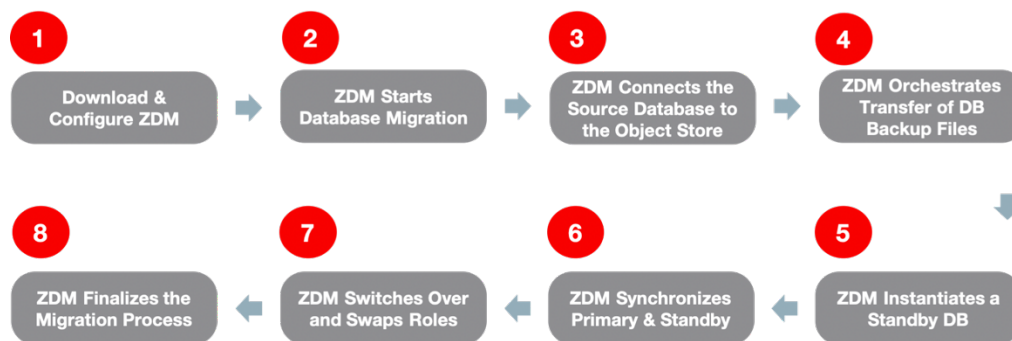


Figure 1 – Oracle Zero Downtime Migration Physical Online Migration Steps

Step 1, Download and Configure ZDM. Step 2, ZDM Starts Database Migration. Step 3 ZDM Connects the Source Database to the Object Store. Step 4, ZDM Orchestrates Transfer of DB Backup Files. Step 5, ZDM Instantiates a Standby Database. Step 6, ZDM Synchronizes Primary and Standby. Step 7, ZDM Switches Over and Swaps Roles. Step 8, ZDM Finalizes the Migration Process.

Supported Platforms and configurations

Oracle ZDM supports the following Oracle Database versions: 11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c, 21c. The source and target databases should be in the same database version. Oracle ZDM supports Oracle Databases hosted on Linux operating systems. The source database can be a single instance database migrating to a single instance or a RAC database, or it can also be a RAC One Node / RAC database, migrating to a RAC database.

Oracle ZDM supports Enterprise & Standard Edition Oracle Databases as source databases. Enterprise Edition Databases are migrated leveraging Oracle Data Guard; Standard Edition Databases are migrated in an offline manner using a backup and restore methodology. Oracle ZDM allows for the source database to be a non-CDB or a container database (CDB) with one or more Pluggable Databases (PDBs). Starting in 21c, ZDM allows for non-CDB to CDB migration with both its physical and logical migration workflows.

ZDM supports on-premises databases to be migrated to:

1. Oracle Database Cloud Service Bare Metal
2. Oracle Database Cloud Service Virtual Machine
3. Exadata Cloud Service, Exadata Cloud at Customer, Exadata On-Premises
4. Autonomous Database (Logical Workflow only)

ZDM Supports the following backup mediums: OCI Object Storage, Oracle Zero Data Loss Recovery Appliance, NFS Storage.

REQUIREMENTS

Zero Downtime Migration Service Host Requirements

Oracle Zero Downtime Migration binaries must be installed on a separate host which fulfils the following requirements:

- Linux host running on Oracle 7
- 100 GB of free storage space
- A `zdm` group and a `zdmuser` as part of this group, please create them as follows:
 - `[root@zdm-servicenode ~]# groupadd zdm -g 1001`
 - `[root@zdm-servicenode ~]# useradd zdmuser -g 1001`
- Following packages must be installed:
 - `glibc-devel`
 - `expect`
 - `unzip`
 - `libaio`
 - `oraclelinux-developer-release-el7`
- All host names and IP address to be used must be present as entries at `/etc/hosts`
- Appropriate permission for source and database access (i.e. ssh keys)

Source Database Requirements

ZDM supports Oracle Database 11g release 2 (11.2.0.4) or later versions, it is important that both the source and target are at the same release version.

- Source database must be in `archive log` mode
- If the source database is on 12c Release 2 and later and Transparent Data Encryption is not enabled:
 - You must configure the Transparent Data Encryption (TDE) Wallet. Guidance on how to configure the wallet can be found at “Setting Up the Transparent Data Encryption Wallet”³
 - Upon setting up the TDE Wallet please ensure that:
 - `WALLET_TYPE` is set to `AUTOLOGIN` or `PASSWORD`
 - Wallet `STATUS` is `OPEN`
 - Wallet must be open on all pluggable databases if the source database is a container database. The master key must be set for all the pluggable databases and the container database.
- Source databases can be Oracle RAC databases, if this is the case, `SNAPSHOT CONTROLFILE` must be configured to point to a shared location on all cluster nodes.

³ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/>

- SCAN listener ports on the source database server and the target database server must allow incoming connections from each other.
- The parameter DB_NAME must be the same on both the source database and the target database.
- The parameter DB_UNIQUE_NAME must be different on both the source database and the target database, this way Oracle Data Guard can identify that the source and target are different.
- The parameter SYSPASSWORD must be the same on both the source database and the target database.
- The Database version must match on both the source database and the target database.

Target Database Requirements

- The target database must be created prior to the migration, this guide will explain this process on a further section.
- The target database version should be the same as the source database version. The target database patch level should also be the same as (or higher than) the source database, if the target database is at a higher patch level than the source database, the customer must run datapatch after the database migration has taken place.
- For Oracle RAC Databases, it is important to setup SSH connectivity between the Oracle RAC nodes for the oracle user.
- Sizing of the target database must be taken into consideration to avoid any shortfalls during the migration. As a rule of thumb, same or larger size than the source database is recommended.
- The parameter DB_NAME must be the same on both the source database and the target database.
- The parameter DB_UNIQUE_NAME must be different on both the source database and the target database, this way Oracle Data Guard can identify that the source and target are different.
- Automatic backups must be disabled for the target database, this is done differently depending on the target cloud platform:
 - OCI DBCS BM/VM: upon creation of the target database there is an option named “Enable automatic backups”, do not select it.
 - Exadata Cloud Service: upon creation of the target database there is an option named “Enable automatic backups”, do not select it.
 - Exadata Cloud at Customer: among the database options there is a section named: “Configure Backups”, go to it, and on the option “Backup Destination” select the “Type” sub-option to None.
- The target database must have Transparent Data Encryption Enabled (TDE). Also, the wallet status must be open, and the wallet type must be either autologin or password.

Backup requirements

- Ensure both the source database server and the target database server can access the backup medium (Object Store for DBCS BM/VM and ExaCS, Recovery Appliance or NFS Storage for ExaCC).
- Sizing must be taken into consideration, so that the source database backups fit without an issue in the backup medium.

Oracle NET Services (SQL*Net) Connectivity Requirements

- Please ensure that all database ports and all SCAN ports are open on both the source database server and the target database server.

Port Requirements

- ZDM Service Node
 - Port 22 must be open, this port is going to be used for SSH, enabling connectivity between the service node and the source database server and the service node and the target database server.
- Source Database Server
 - Port 1521 must be open and not blocked by a firewall, this port will be used for Oracle NET Services connectivity between the source database server and target database server. This connectivity will enable proper Data Guard Synchronization.
 - Port 443 must be open and not blocked by a firewall, this port will be used to access the Object Store.
- Target Database Server
 - Port 1521 must be open and not blocked by a firewall, this port will be used for Oracle NET Services connectivity between the source database server and target database server. This connectivity will enable proper Data Guard Synchronization.
 - Port 443 must be open and not blocked by a firewall, this port will be used to access the Object Store.

ZDM INSTALLATION & PRE-MIGRATION CONFIGURATION

Oracle ZDM Service Node Configuration

Visit www.oracle.com/database/technologies/rac/zdm-downloads.html

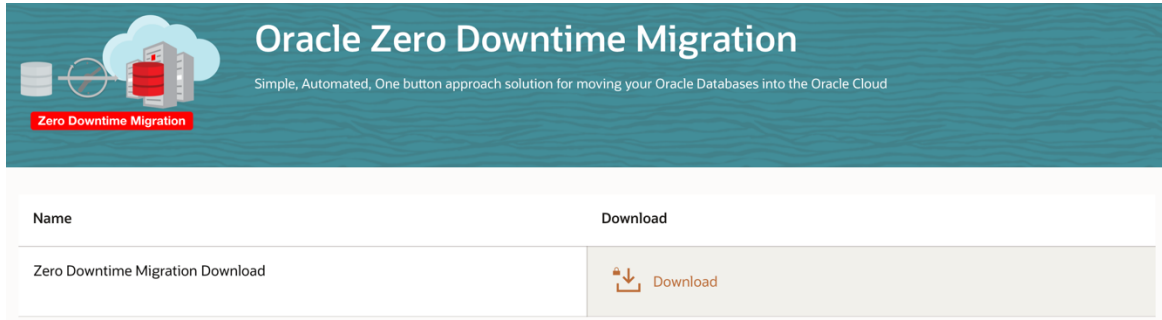


Figure 2 – “Oracle Zero Downtime Migration Downloads” Screenshot from www.oracle.com/goto/zdm

Accept the Oracle License Agreement <https://www.oracle.com/technetwork/licenses/sqldev-license-152021.html#licenseContent> and download the ZDM binaries on the ZDM Service host. Upon download, switch to the previously created `zdmuser` user. Unzip the binaries using the `unzip` command. Remember that the ZDM service node runs on a separate host, the best practice for the migration looks as per the figure below:

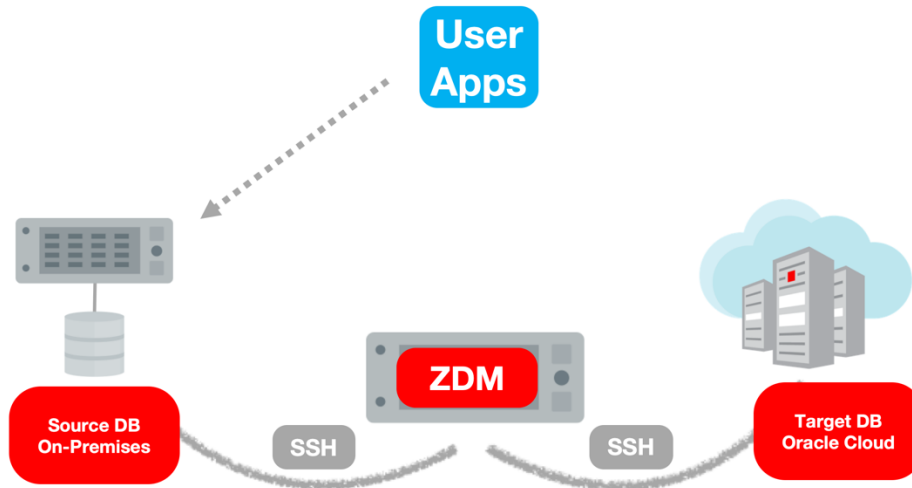


Figure 3 – Oracle Zero Downtime Migration Physical Online Migration Architecture. ZDM Service Node connect to Source Database On-Premises and Target Database in the Oracle Cloud via SSH. User Applications connected to the Source Database On-Premises.

Upon extracting the binaries, create the following directories:

- `oraclehome`: directory where ZDM will be installed
- `oraclebase`: directory where all ZDM’s configuration files and logs will be stored

Install the required packages, `glibc-devel`, `expect`, etc:

- `[root@zdm-servicenode ~]# yum install glibc-devel*`
- `[root@zdm-servicenode ~]# yum install expect`

Proceed to execute ZDM's installation script `zdmuser`:

```
./zdminstall.sh setup oraclehome=zdm_oracle_home oraclebase=zdm_base_directory ziploc=zdm_loc  
-zdm
```

Bear in mind that `ziploc` refers to the full path `zdm_home.zip` is located. This file is part of the uncompressed bundle from the downloadable binaries from www.oracle.com/goto/zdm

Due to ZDM inherent relationship with Grid Infrastructure, a set of messages requiring scripts to be ran as a root user will appear upon installation completion. Please disregard this messages, ZDM's installation process does not require any actions as root. Furthermore, as shown below, the installer will request for certain scripts to be run as root, please disregard this.

If you have already installed ZDM and want to perform an upgrade, please visit the latest product documentation, accessible from our product page www.oracle.com/goto/zdm for a step by step guide on how to upgrate ZDM.

Upon successful installation and before performing any Oracle Database migration, the ZDM service must be started and installation success must be verified. In order to so, please execute the following as `zdmuser`:

```
/zdm_oracle_home/bin/zdmservice start  
/zdm_oracle_home/bin/zdmservice status
```

The `zdmservice start` command will start the `zdmservice`. The `zdmservice status` command should return the service status and show `running:true`. This guarantees that the `zdmservice` was properly installed and it is up and running. If the installation process failed or there are errors present, please refer to the Frequently Asked Questions section at the end of this guide.

```

[zdmuser@zdm-servicenode media]$ ./zdminstall.sh setup oraclehome=/u01/app/oracle/zdmhome
oraclebase=/u01/app/oracle/zdmbase ziploc=/u01/media/zdm_home.zip -zdm
-----
Unzipping shiphome to gridhome
-----
Unzipping shiphome...

Shiphome unzipped successfully..
-----
##### Starting GridHome Software Only Installation #####
-----
Launching Oracle Grid Infrastructure Setup Wizard...

A WARNING WILL BE DISPLAYED HERE, PLEASE DISREGARD

The response file for this session can be found at:
/u01/app/oracle/zdmhome/install/response/grid_.rsp

You can find the log of this install session at:
/tmp/GridSetupActions/gridSetupActions.log

As a root user, execute the following script(s):
    1. /u01/app/oracle/zdmhome/inventory/orainstRoot.sh
    2. /u01/app/oracle/zdmhome/root.sh

Execute /u01/app/oracle/zdmhome/inventory/orainstRoot.sh on the following nodes:
[zdm-servicenode]
Execute /u01/app/oracle/zdmhome/root.sh on the following nodes:
[zdm-servicenode]

Successfully Setup Software with warning(s).
Moved the install session logs to:
/u01/app/oracle/zdmhome/inventory/logs/GridSetupActions
=====
PLEASE IGNORE THE MESSAGE ABOVE TO RUN SCRIPT(S) AS ROOT USER.
ZDM INSTALL DOES NOT NEED ANY ACTIONS AS ROOT.
=====

making dir /u01/app/oracle/zdmbase/crsdata/zdm-servicenode/rhp/conf
-----
Generating Preference file
-----
/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/rhp/conf/rhp.pref
-----
Generating Root Certificate
-----
Cluster root certificate generated successfully.
-----
Generating CA CERTS file
-----
spawn /u01/app/oracle/zdmhome/bin/crskeytoolctl -copycacerts -filestore
/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/security/
Enter JRE cacerts truststore password:
JRE cacerts copied to file [/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/security/cacerts].
-----
Generating nogi.enabled file
-----
nogi.enabled file generated successfully
-----
Generating standalone_config.properties file
-----
Setting base folder permissions
-----
Copying service script to bin folder in Oracle Home
ZDM setup without GI finished successfully...

```

```

[zdmuser@zdm-servicenode bin]$ ./zdm-service start
No instance detected, starting zdm-service

[jwcctl debug] Environment ready to start JWC
[jwcctl debug] Return code of initialization: [0]

[jwcctl debug] ... BEGIN_DEBUG [Action= start] ...
Start JWC
[jwcctl debug] Loading configuration file: /u01/app/oracle/zdmbase/crsdata/zdm-
servicenode/rhp/conf/jwc.properties
[jwcctl debug] oracle.jmx.login.credstore = CRSCRED
[jwcctl debug] oracle.jmx.login.args = DOMAIN=rhp
[jwcctl debug] oracle.rmi.url =
service:jmx:rmi://{0}:{1,number,#}/jndi/rmi://{0}:{1,number,#}/jmxrmi
[jwcctl debug] oracle.http.url = http://{0}:{1,number,#}/rhp/gridhome
[jwcctl debug] oracle.jwc.tls.clientauth = false
[jwcctl debug] oracle.jwc.tls.rmi.clientfactory = RELOADABLE
[jwcctl debug] oracle.jwc.lifecycle.start.log.fileName = JWCStartEvent.log
[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC containers not found ...

[jwcctl debug] Start command:-server -Xms128M -Xmx384M -Djava.awt.headless=true -
Ddisable.checkForUpdate=true -Djava.util.logging.config.file=/u01/app/oracle/zdmbase/crsdata/zdm-
servicenode/rhp/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -DTRACING.ENABLED=true -
DTRACING.LEVEL=2 -Doracle.wlm.dbwlmlogger.logging.le vel=FINEST -Duse_scan_IP=true
-Djava.rmi.server.hostname=zdm-servicenode -Doracle.http.port=8896 -Doracle.jmx.port=8895 -
Doracle.tls.enabled=false -Doracle.jwc.tls.http.enabled=false
Doracle.rhp.storagebase=/u01/app/oracle/zdmbase -Djava.security.egd=file:/dev/urandom -
Doracle.jwc.wallet.path=/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/security -Doracle.jmx
.login.credstore=WALLET -Dcatalina.home=/u01/app/oracle/zdmhome/tomcat -
Dcatalina.base=/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/rhp -
Djava.io.tmpdir=/u01/app/oracle/zdmbase/cr sdata/zdm-servicenode/rhp/temp -
Doracle.home=/u01/app/oracle/zdmhome -Doracle.jwc.mode=STANDALONE -classpath
/u01/app/oracle/zdmhome/jlib/cryptoj.jar:/u01/app/oracle/zdmhome/jlib/ora
clepki.jar:/u01/app/oracle/zdmhome/jlib/osdt_core.jar:/u01/app/oracle/zdmhome/jlib/osdt_cert.jar:/u01
/app/oracle/zdmhome/tomcat/lib/tomcat-juli.jar:/u01/app/oracle/zdmhome/tomcat/lib
/bootstrap.jar:/u01/app/oracle/zdmhome/jlib/jwc-logging.jar org.apache.catalina.startup.Bootstrap
start
[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC Container (pid=26626) ...
[jwcctl debug] ... JWC Container running (pid=26626) ...

```

```
[jwcctl debug] Check command:-Djava.net.preferIPv6Addresses=true -
Dcatalina.base=/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/rhp -
Doracle.wlm.dbwlmlogger.logging.level=FINEST -
Doracle.jwc.client.logger.file.name=/u01/app/oracle/zdmbase/crsdata/zdm-
servicenode/rhp/logs/jwc_checker_stdout_err_%g.log -Doracle.jwc.client.logger.file.number=10 -
Doracle.jwc.client.logger.file.size=1048576 -
Doracle.jwc.wallet.path=/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/security -
Doracle.jmx.login.credstore=WALLET -Doracle.tls.enabled=false -Doracl
e.jwc.tls.http.enabled=false -classpath /u01/app/oracle/zdmhome/jlib/jwc-
logging.jar:/u01/app/oracle/zdmhome/jlib/jwc-security.jar:/u01/app/oracle/zdmhome/jlib/jwc-
client.jar:/u01/ap p/oracle/zdmhome/jlib/jwc-
cred.jar:/u01/app/oracle/zdmhome/jlib/srvm.jar:/u01/app/oracle/zdmhome/jlib/srvmhas.jar:/u01/app/orac
le/zdmhome/jlib/cryptoj.jar:/u01/app/oracle/zdmhome/jli
b/oraclepki.jar:/u01/app/oracle/zdmhome/jlib/osdt_core.jar:/u01/app/oracle/zdmhome/jlib/osdt_cert.jar
:/u01/app/oracle/zdmhome/tomcat/lib/tomcat-juli.jar oracle.cluster.jwc.tomcat.cli
ent.JWCChecker localhost 8896 -1
```

```
[jwcctl debug] ... JWC Container is ready ...
[jwcctl debug] ... START - Return code = 0 ...
[jwcctl debug] ... END_DEBUG [Action=start] ...
[jwcctl debug] Return code of AGENT: [0]
```

```
Return code is 0
Server started successfully.
```

```
[zdmuser@zdm-servicenode bin]$
```

You can verify that the ZDM service was setup successfully by trying to query an image, in order to so, please execute the following as zdmuser:

```
./zdm_oracle_home/bin/zdmcli query image
```

```
[zdmuser@tool bin]$ ./zdmcli query image
x.x.x.x.x.host.com: Audit ID: 2
No image has been configured
```

Configuring Connectivity between ZDM Service Node and Source and Target Database Servers

The ZDM service node must connect securely to the source database server and the target database server to ensure a swift migration process. In order to do so, ssh authentication key pairs must be available (without a passphrase) for the recently created `zdmuser`. In case the keys are not present or they require a passphrase, you can generate them following our **“Generating a Private SSH Key Without a Passphrase”** section in our documentation.

<https://docs.oracle.com/en/database/oracle/zero-downtime-migration/>

Upon creation of the private ssh key files, the following files will be available: `id_rsa` and `id_rsa.pub` files. It is recommended as a best practice to rename the `id_rsa.pub` file to `<zdm_service_node_name>.ppk`. Finally the public keys must be added to the source database server and the target database server (`opc_user_home/.ssh/authorized_keys`) files.

Edit the ZDM service node `/etc/hosts` file, adding the source and target database servers hostname and IP information:

```
#Source
111.1.1.1 source.node-1234 source

#OCI Target
129.1.2.3 target.node-1234-oci target
```

Please bear in mind that the alias of both source and target should be the name without the domain information. Finally, test the connectivity from the ZDM Service node to the source database server and target database server, using the `ssh -i` command.

```
[zdmuser@zdm-servicenode ~]# ssh -i zdm_service_tool.ppk opc@source
[zdmuser@zdm-servicenode ~]# ssh -i zdm_service_tool.ppk opc@target

[zdmuser@tool .ssh]$ ssh -i zdm_service_tool.ppk opc@source
The authenticity of host 'source (X.X.X.X)' can't be established.
RSA key fingerprint is .....
RSA key fingerprint is .....
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'source,X.X.X.X' (RSA) to the list of known hosts.
-- WARNING -- This system is for the use of authorized users only. Individuals
using this computer system without authority or in excess of their authority
are subject to having all their activities on this system monitored and
recorded by system personnel. Anyone using this system expressly consents to
such monitoring and is advised that if such monitoring reveals possible
evidence of criminal activity system personnel may provide the evidence of such
monitoring to law enforcement officials.

Last login: from inet-x-x-x-x.x.com
*** ALERTS::INFO BEGIN ***
ALERTS:: System Alert: [Database User accounts check Failed]
ALERTS:: Please check /var/opt/oracle/log/monitor_alerts/monitor_alerts.log
*** ALERTS::INFO END ***
[opc@source ~]$ exit
```

```
[zdmuser@tool .ssh]$ ssh -i zdm_service_tool.ppk opc@target
The authenticity of host 'target (x.x.x.x)' can't be established.
ECDSA key fingerprint is .....
ECDSA key fingerprint is .....
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'target,x.x.x.x' (ECDSA) to the list of known hosts.
Last login: from x.x.x.x
[opc@target ~]$ exit
```

Configuring Connectivity between the Source and Target Database Servers

The source database server and the target database server must be able to connect via Oracle NET Services (SQL*net) in order to guarantee Data Guard operations. There are two options for connectivity between source and target:

1. The first option to ensure connectivity can be over SCAN, through the source/target SCAN port which must be resolvable from both the source database server and the target database server.
2. The second option to ensure connectivity requires setting up an SSH tunnel from the source database server to the target database server.

Option 1 - SCAN Connectivity

In order to guarantee connectivity via SCAN between the source and target database servers, both `/etc/hosts` files must be edited.

Source database server `/etc/hosts` file entry to be added:

```
#Source
111.1.1.1 source.node-1234-oci source
111.1.1.1 source-priv.node-1234-oci source-priv
111.1.1.1 source-vip.node-1234-oci source-vip
111.1.1.1 source-scan.node-1234-oci source-scan
#Target
129.1.2.3 target.node-1234-oci target
129.1.2.3 target-scan.node-1234-oci target-scan
```

Target database server `/etc/hosts` file entry to be added:

```
#Target
129.1.2.3 target.node-1234-oci target
129.1.2.3 target-priv.node-1234-oci target-priv
129.1.2.3 target-vip.node-1234-oci target-vip
129.1.2.3 target-scan.node-1234-oci target-scan
#Source
111.1.1.1 source.node-1234 source
111.1.1.1 source-scan.node-1234 source-scan
```

In the case of a source database that is not Grid Infrastructure based, then, the source scan entry in the `/etc/hosts` target file is not required.

Bear in mind that some of these entries might already be on the `/etc/hosts` files by default, this is just a sample of how the files should look for ZDM to work properly.

Afterwards, connectivity can be tested via the `tnsping` utility, trying to reach the target database server from the source database server and viceversa via port 1521.

```
[oracle@source ~]$ tnsping target:1521
```

```
[oracle@target ~]$ tnsping source:1521
```

Option 2 - SSH Tunnel

In order to setup an SSH tunnel between the source and target database server, please follow the steps outlined on ZDM's product documentation section "**Configuring Connectivity between the Source and Target Database Servers**" subsection "**Option 2**".

<https://docs.oracle.com/en/database/oracle/zero-downtime-migration/>

Creating a Target Database

In order to create a placeholder target database, login to your Oracle Cloud account, and access the DB systems tab by clicking on the **Bare Metal, VM and Exadata menu**.



Figure 4 – “Bare Metal, VM and Exadata Menu” screenshot from Oracle Cloud.

Click on the **Create DB System** option.

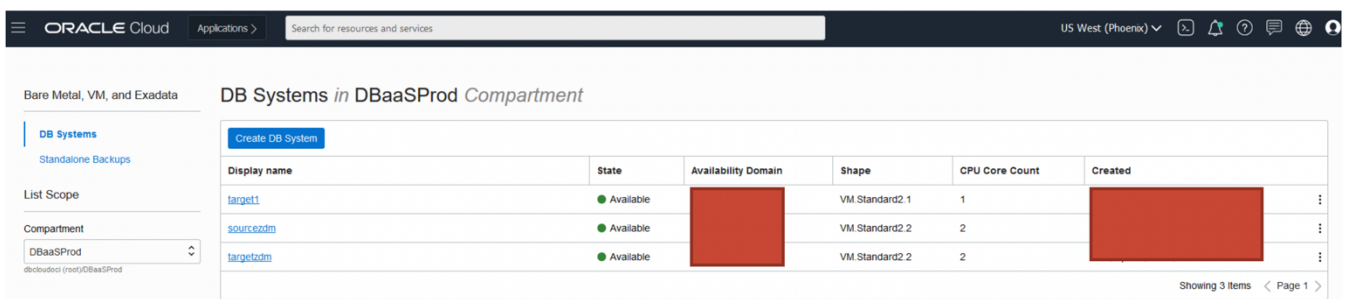


Figure 5 – “Create DB System” screenshot from Oracle Cloud.

Select the appropriate values regarding: Compartment, DB System name, Availability domain and Shape.

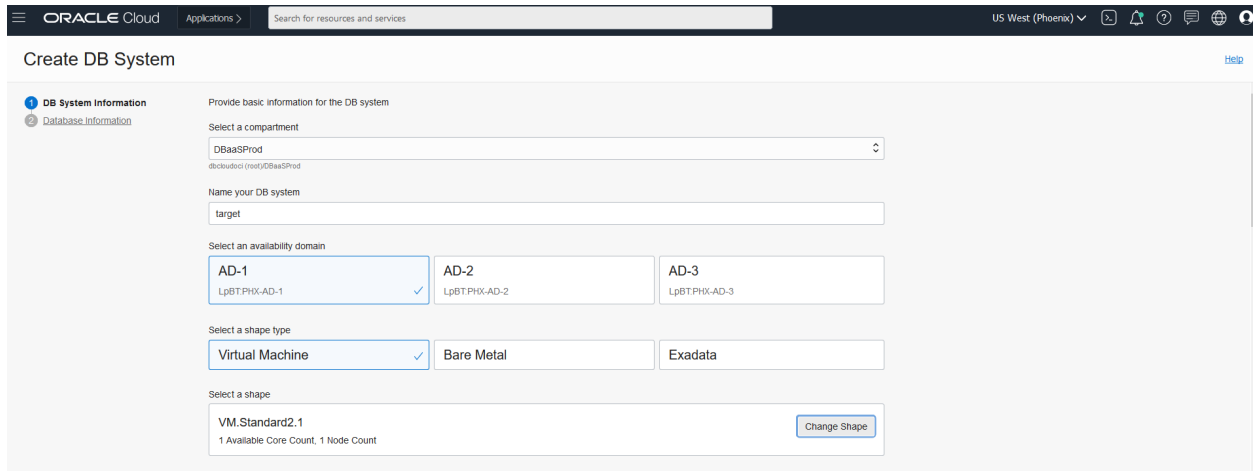


Figure 6 – “Create DB System” window screenshot, with DB System Information Options from Oracle Cloud.

Configure the DB system by selecting: Total node count and Oracle Database software edition. Choose the Storage Management Software, bear in mind that Oracle ZDM only supports migrations to Databases which use Oracle Grid Infrastructure. Logical Volume Manager is not supported.

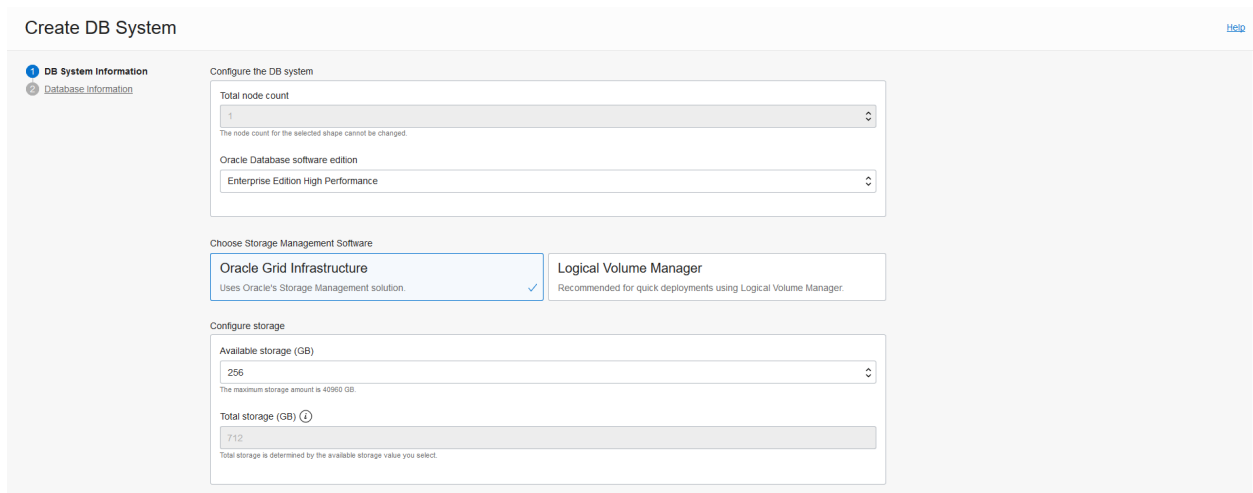


Figure 7 – “Create DB System” window screenshot, with DB System, Storage Management Software and Storage configuration options from Oracle Cloud.

Select **Upload SSH Keys** and Choose a license type.

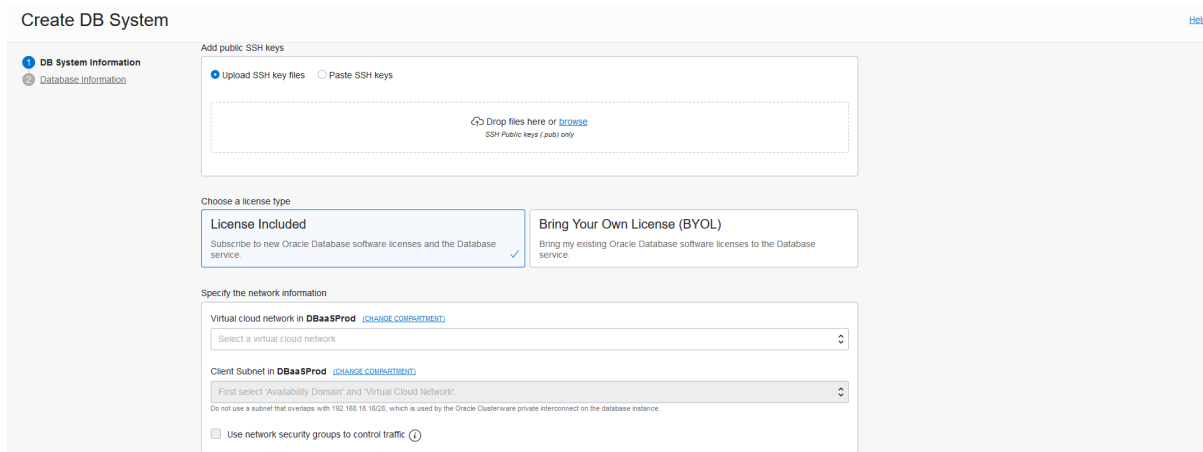


Figure 8 – “Create DB System” window screenshot, with SSH Keys option from Oracle Cloud.

Specify the network information according to your setup, this includes Hostname prefix, Host domain name and Host and domain URL, this last value is autogenerated and determined by the prefix and the host domain name.

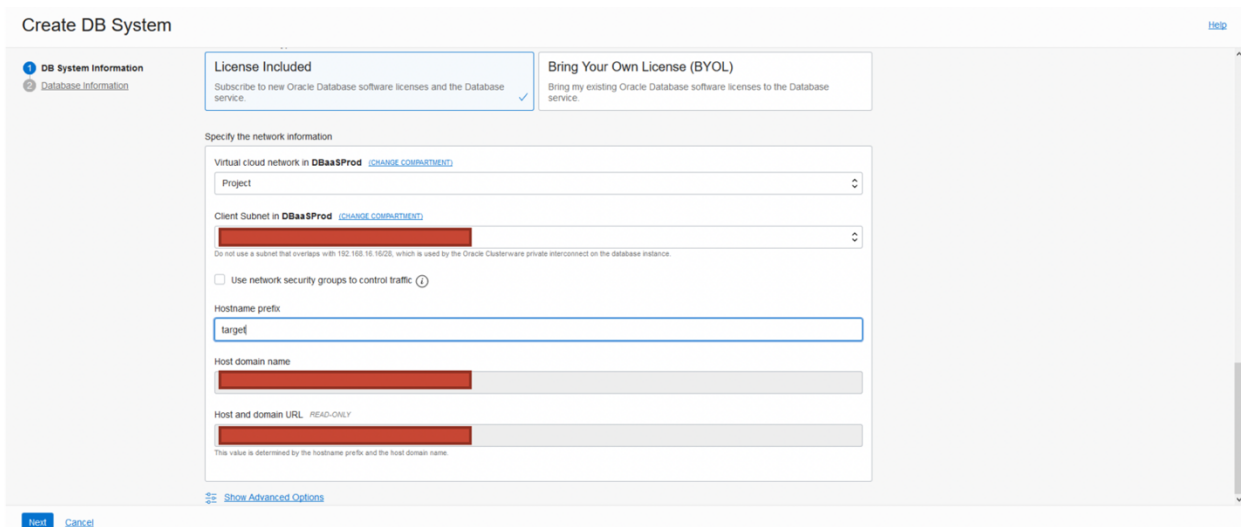


Figure 9 – “Create DB System” window screenshot, with Network configuration information from Oracle Cloud.

Proceed to provide information related to the target database, this includes: **Database name**, **Database version**, and the **administrator credentials**. Please bear in mind that **Database name** value should be the same as the source database name.

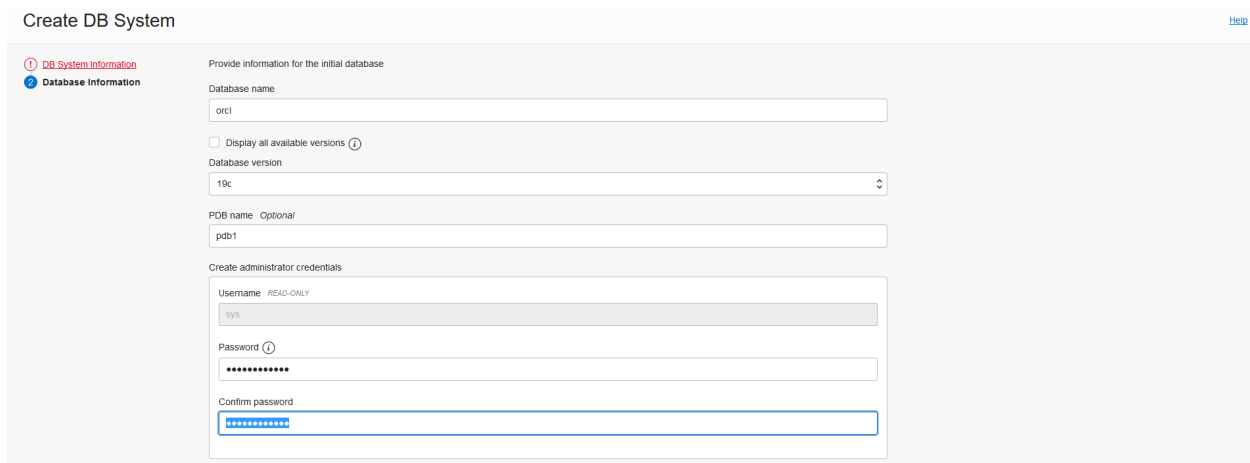
The screenshot shows the 'Create DB System' window with the 'Database Information' step selected. The form includes fields for 'Database name' (value: orcl), 'Database version' (value: 19c), and 'PDB name' (value: pdb1). Under 'Create administrator credentials', there are fields for 'Username' (value: sys), 'Password', and 'Confirm password'. A 'Help' link is visible in the top right corner.

Figure 10 – “Create DB System” window screenshot, with Database configuration information from Oracle Cloud.

Finally, select **Transaction Processing** as the database workload type and if required change your character set to the desired character set. When you are ready, click on **Create DB System**.

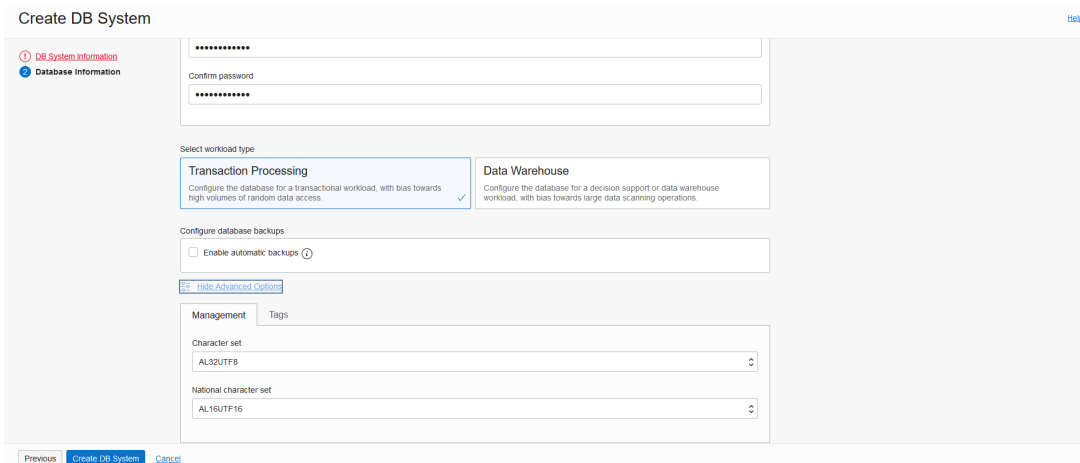
The screenshot shows the 'Create DB System' window with the 'Workload Type' section. The 'Transaction Processing' option is selected, with a description: 'Configure the database for a transactional workload, with bias towards high volumes of random data access.' The 'Data Warehouse' option is also visible. Below, there are options for 'Configure database backups' (with 'Enable automatic backups' unchecked) and 'Advanced Options' (with 'Management' selected). Under 'Management', the 'Character set' is set to 'AL32UTF8' and the 'National character set' is set to 'AL16UTF16'. Navigation buttons 'Previous', 'Create DB System', and 'Cancel' are at the bottom.

Figure 11 – “Create DB System” window screenshot, with Workload Type information from Oracle Cloud.

Setting up a Transparent Data Encryption Wallet

Oracle Cloud encrypts all of its databases by default, to do so, Oracle uses its Transparent Data Encryption (TDE) technology. This means, that when a customer migrates a database to the Oracle Cloud, even if the source database was not encrypted, the target database will be encrypted with TDE.

Oracle Zero Downtime Migration supports unencrypted databases as source databases, taking care of the target database encryption during the migration process. Since not all Oracle source databases are encrypted by default, there are several steps that must be taken, specifically if the source database is on 12c release 2 or later. In such cases, a Transparent Data Encryption wallet must be configured prior to performing the migration.

Please visit Oracle Zero Downtime Migration Documentation’s *Setting Up the Transparent Data Encryption Wallet* section, for a step by step guide on how to perform this task:

<https://docs.oracle.com/en/database/oracle/zero-downtime-migration/>

Creating Object Storage for Database Backup

Oracle Zero Downtime migration leverages the Object Storage to place the RMAN generated backups of the source database. This backups are transferred from the source database to the object storage via https connectivity. Once ZDM completes this process, it instantiates a standby database leveraging this backup files. Please follow this steps to create an Object Storage for Database Backup.

Login to your Oracle Cloud Account and click on the Object Storage menu and the Object Storage option.

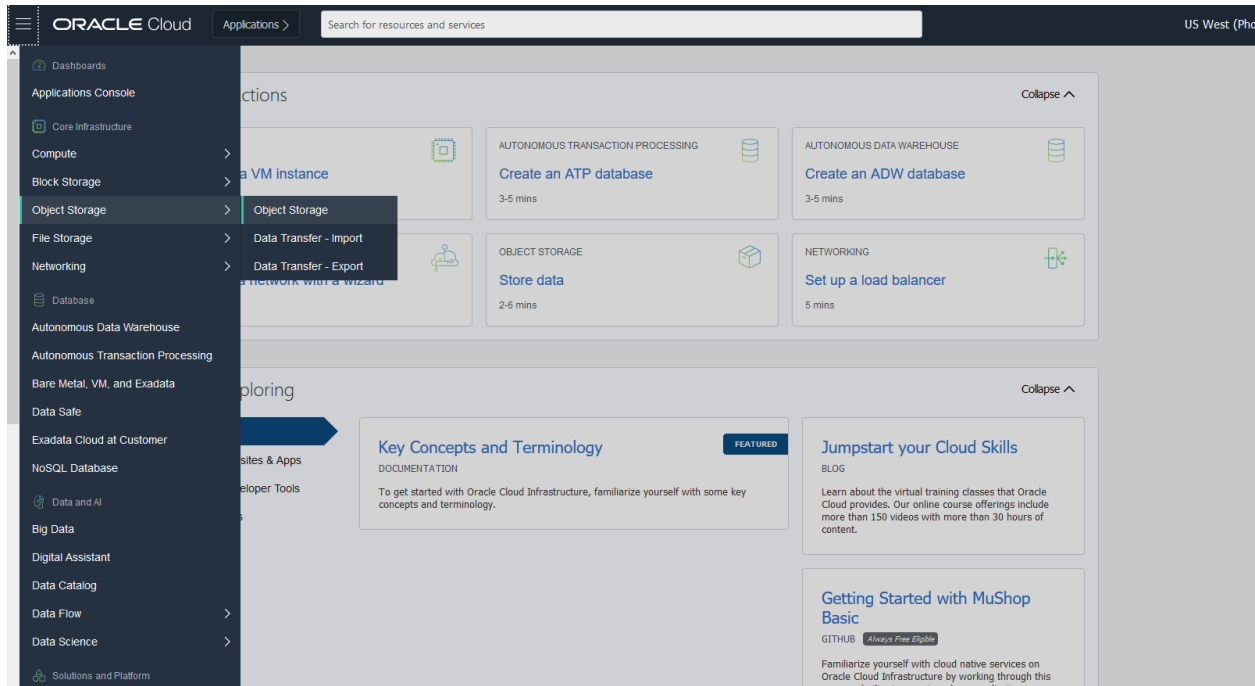


Figure 12 – “Object Storage” window screenshot from Oracle Cloud.

Select your compartment and click on the **Create Bucket** option. It is important to bear in mind that the compartment **must** be in the same local region where you have created the target placeholder database.

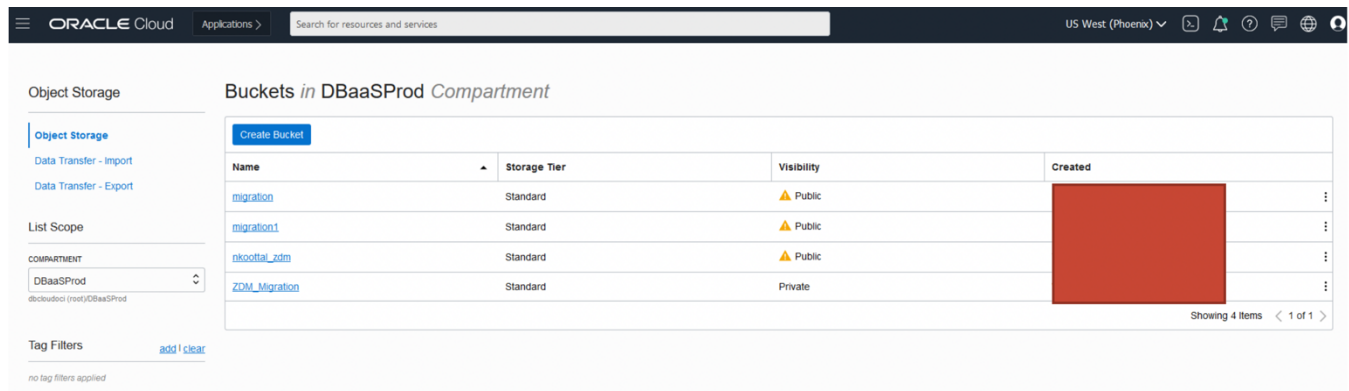


Figure 13 – “Create Bucket” window screenshot from Oracle Cloud.

Enter a desired Bucket Name, select the Standard for the Storage Tier and Select Encryption using Oracle Managed Keys. Finally, click on the **Create Bucket** option.

Create Bucket [Help](#) [Cancel](#)

BUCKET NAME

ZDM_Migration

STORAGE TIER

Storage tier for a bucket can only be specified during creation. Once set, you cannot change the storage tier in which a bucket resides.

STANDARD

ARCHIVE

OBJECT EVENTS ⓘ

EMIT OBJECT EVENTS

OBJECT VERSIONING ⓘ

ENABLE OBJECT VERSIONING

ENCRYPTION

ENCRYPT USING ORACLE MANAGED KEYS
Leaves all encryption-related matters to Oracle.

ENCRYPT USING CUSTOMER-MANAGED KEYS
Requires you to have access to a valid Key Management key. [Learn More](#)

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.

[Learn more about tagging](#)

TAG NAMESPACE TAG KEY VALUE

None (add a free-form tag) [] [] ×

+ Additional Tag

Create Bucket Cancel

Figure 14 – “Create Bucket” window screenshot with Bucket Configuration information from Oracle Cloud.

Generating an Authentication Token

Login to your Oracle Cloud Account, click on the **Identity** option and click on the **Users** option.

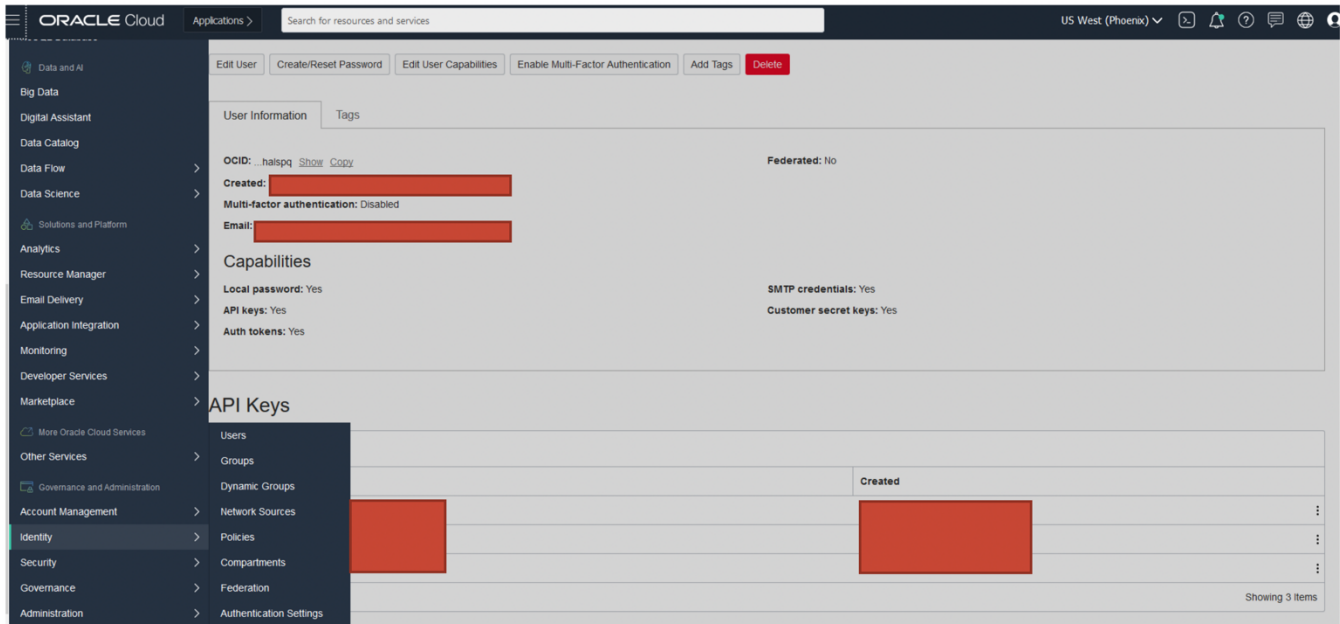


Figure 15 – “Identity > Users” window screenshot from Oracle Cloud.

Select the user you want to use for Object Storage authentication, click the **Auth Tokens** option and click the **Generate Token** option.

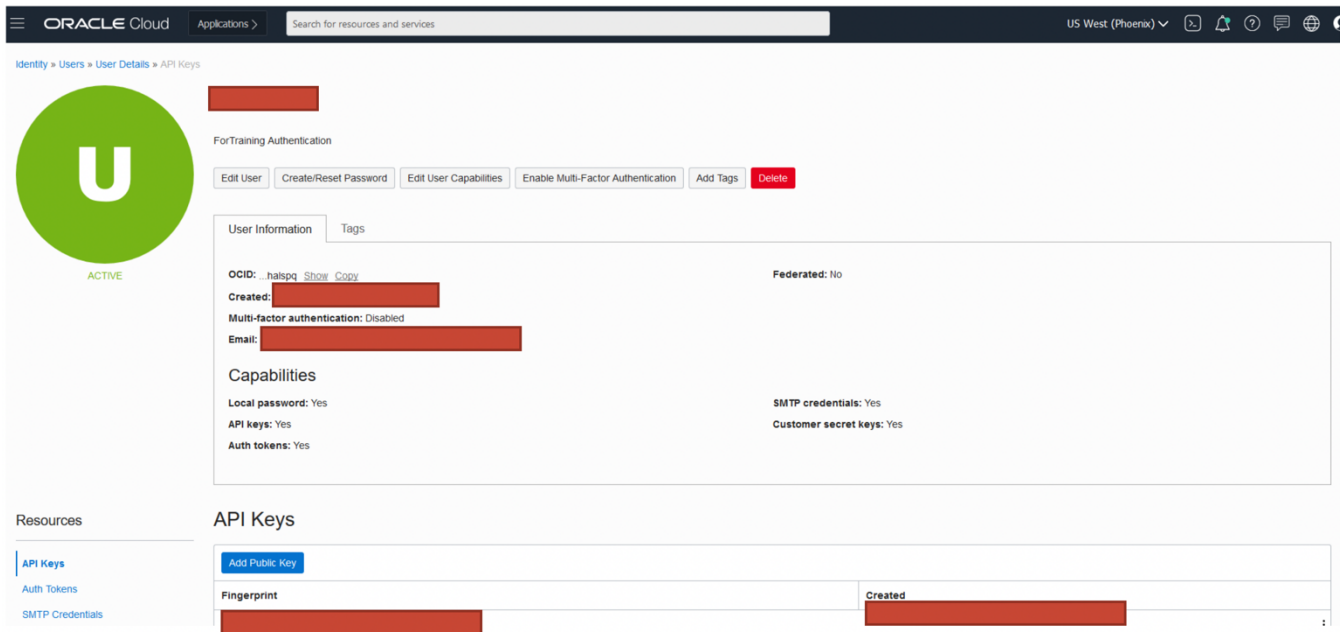


Figure 16 – “Auth Token” window screenshot from Oracle Cloud.

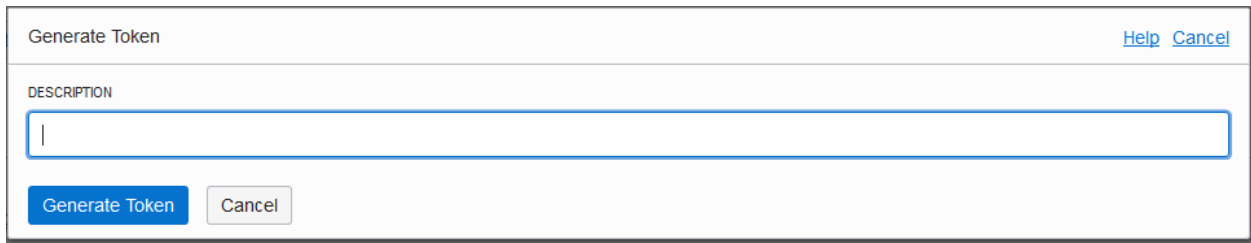


Figure 17 – “Auth Token” window screenshot from Oracle Cloud.

Bear in mind that it is important to copy the token when prompted to do so, the token will not afterwards. If you do not copy it now, you will need to create a new one later.

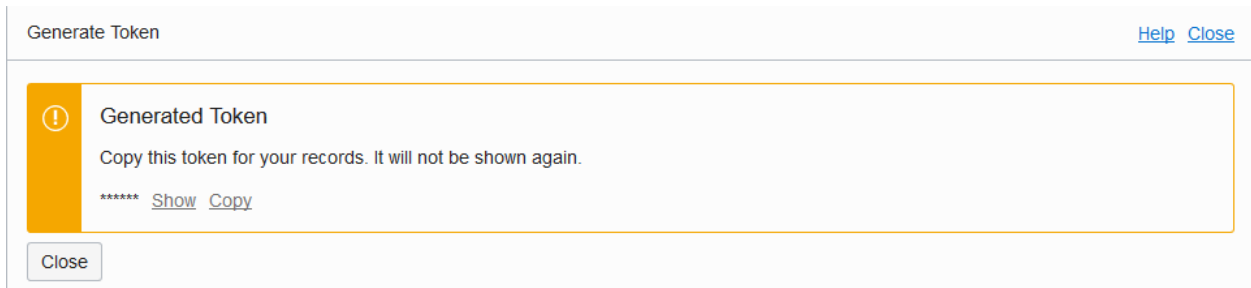


Figure 18 – “Generate Token” window screenshot from Oracle Cloud.

Preparing the Response File

Response File Parameters

PARAMETER	DESCRIPTION
TGT_DB_UNIQUE_NAME	Let's the ZDM service node know the target database db_unique_name value
MIGRATION_METHOD	<ul style="list-style-type: none">• ONLINE_PHYSICAL → ZDM will leverage Data Guard for the migration process.• OFFLINE_PHYSICAL → ZDM will use a backup and restore methodology for the migration.
PLATFORM_TYPE	<p>Specifies the target platform for the migration process</p> <ul style="list-style-type: none">• VMDB → this option is valid for Virtual Machine or Bare Metal.• EXACC → this option is valid for Exadata Cloud at Customer.• EXACS → this option is valid for Exadata Cloud Service.• NON_CLOUD → this option is valid for Exadata On-Premises.
SRC_HTTP_PROXY_URL	Specifies ssh proxy url for the source database server in case it is required
SRC_HTTP_PROXY_PORT	Specifies ssh proxy port for the source database server in case it is required
SRC_CONFIG_LOCATION	Specifies the ssh configuration file location for the source database server in case it is not present in the default location: <user_home>/.ssh/config
SRC_TIMEZONE	Timezone value of the source database, only required for single instance source databases where Grid Infrastructure is not configured.
SRC_OSS_PROXY_HOST	Specifies proxy url for the source database server in case it is required to connect to the object store
SRC_OSS_PROXY_PORT	Specifies proxy port for the source database server in case it is required to connect to the object store
TGT_HTTP_PROXY_URL	Specifies ssh proxy url for the target database server in case it is required
TGT_HTTP_PROXY_PORT	Specifies ssh proxy port for the target database server in case it is required

TGT_CONFIG_LOCATION	Specifies the ssh configuration file location for the target database server in case it is not present in the default location: <user_home>/.ssh/config
TGT_SSH_TUNNEL_PORT	In the case of tunneling configuration being present, this parameter specifies the port where the ssh tunnels between source and target for the setup of the Oracle NET Services (SQL * Net) connectivity
TGT_OSS_PROXY_HOST	Specifies proxy url for the target database server in case it is required to connect to the object store
TGT_OSS_PROXY_PORT	Specifies proxy port for the source database server in case it is required to connect to the object store
TGT_DATADG, TGT_REDODG, TGT_RECODG	In case the target database is using ASM for datafiles this value specifies the database data files properties.
TGT_DATAACFS, TGT_REDOACFS, TGT_RECOACFS	In case the target database is using ACFS for datafiles this value specifies the database data files properties.
HOST	Specifies the Object Storage Namespace, this value can be found under the cloud console menu > administration > tenancy detail. The tenancy information section show the valid url for the object storage namespace
OPC_CONTAINER	Object storage bucket name
SRC_ZDLRA_WALLET_LOC	Wallet location in the source database server when the Zero Data Loss Recovery Appliance is used as backup medium.
TGT_ZDLRA_WALLET_LOC	Wallet location in the target database server when the Zero Data Loss Recovery Appliance is used as backup medium.
ZDLRA_CRED_ALIAS	Wallet credential alias when the Zero Data Loss Recovery Appliance is used as backup medium
BACKUP_PATH	Allows ZDM to know where the backup location is located <ul style="list-style-type: none"> • STORAGEPATH • EXTBACKUP
SKIP_FALLBACK	Specifies if redo logs from the target database are not be sent to the source database upon role swap. The default value is FALSE.
TGT_RETAIN_DB_UNIQUE_NAME	Specifies whether to ship redo logs from Oracle Cloud to the on-premises standby, observe the environment for some time, and remove the fallback late

SHUTDOWN_SRC	Specifies if the source database is to be shutdown after the migration, if the value is set to TRUE it performs the latter, if it is set to FALSE it does not.
SRC_RMAN_CHANNELS	Specifies the number of Recovery Manager Channels at the source database server used to perform RMAN backups, default value is 10
TGT_RMAN_CHANNELS	Specifies the number of Recovery Manager Channels at the target database server used to perform RMAN backups, default value is 10
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL	specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_FULL_SRC migration job phase.
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL	specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_INCREMENTAL_SRC migration job phase.
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL	specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_DIFFERENTIAL_SRC migration job phase.
ZDM_CLONE_TGT_MONITORING_INTERVAL	specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_CLONE_TGT migration job phase.
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL	specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RECOVER_TGT migration job phase.
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL	specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RESTORE_TGT migration job phase.
SKIP_SRC_SERVICE_RETENTION	lets the user skip source database service retention. Default value is FALSE.
ZDM_BACKUP_RETENTION_WINDOW	lets the user specify the number of days after which database backups created by Zero Downtime Migration become obsolete. The default is 60 (days).
ZDM_OPC_RETRY_WAIT_TIME	lets the user specify the Object Store retry wait time in seconds. The default value is 529 (seconds).
ZDM_OPC_RETRY_COUNT	lets the user specify how many times to retry the Object Store. The default value is 0 (zero).
ZDM_CURL_LOCATION	lets the user specify a custom location for CURL binary on the source. The default value is /usr/bin/curl.
ZDM_LOG_OSS_PAR_URL	lets the user specify a pre-authenticated URL to which to upload log files. The logs also capture the current phase as well as the execution status of the phase.

ZDM_SRC_TNS_ADMIN	lets the user specify a custom location for TNS_ADMIN on the source database when there is no Grid Infrastructure.
SRC_SSH_RETRY_TIMEOUT	Specifies a timeout value in minutes for ZDM to stop attempting SSH connections to the Source
TGT_SSH_RETRY_TIMEOUT	Specifies a timeout value in minutes for ZDM to stop attempting SSH connections to the Target
TGT_SKIP_DATAPATCH	lets the user specify whether or not Zero Downtime Migration runs the datapatch utility on the target database as part of the post-migration tasks. The default value is FALSE, which allows Zero Downtime Migration to run datapatch.
MAX_DATAPATCH_DURATION_MINS	lets the user configure a timeout value, in minutes, after which, if the datapatch utility has failed to complete, then the operation is stopped. There is no default value. By default, Zero Downtime Migration waits until datapatch completes.
DATAPATCH_WITH_ONE_INSTANCE_RUNNING	when set to TRUE, lets Zero Downtime Migration stop all instances except one running on the target database server when the datapatch utility runs. When datapatch completes, all of the stopped instances are started. The default value is FALSE.
NON_CDBTOPDB_CONVERSION	Specifies if a non-cdb to cdb conversion must take place as part of the migration process. The default value is FALSE.

DATABASE MIGRATION WITH ZDM

Performing a Database Migration pre-check process

To perform a database migration pre-check process, the `zdmcli migrate database` command with the `-eval` flag must be executed. For example, this is how the command would look like:

```
./zdmcli migrate database -sourcedb db1_phx -sourcenode sourcezdm -srcauth zdmauth  
-srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk  
-srcarg3 sudo_location:/usr/bin/sudo -targetnode targetzdm  
-targethome /u01/app/oracle/product/18.01  
-backupuser bkpuser -rsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtarg1 user:opc  
-tgtarg2 identity_file:/home/zdmuser/.ssh/zdm-spk -tgtarg3 sudo_location:/usr/bin/sudo  
-ignore ALL -eval
```

FLAG	DESCRIPTION
-sourcedb or -sourcesid	Source database name, in this case the database name is <code>db1_phx</code> If the source database is a single instance database deployed on a non Grid Infrastructure environment, the correct flag should be <code>-sourcesid</code> alongside the database name.
-sourcenode	Source database server host name, in this case it is <code>sourcezdm</code>
-srcauth	Authentication plug-in name required to access the source database server. This let's ZDM know that a number n of plugin arguments are to come, specifying the specific authentications to be followed.
-srcarg1	Plugin argument, in this case it specifies the <code>opc</code> user
-srcarg2	Plugin argument, in this case it specifies the location of the public keys: <code>/home/zdmuser/.ssh/zdm-servicenode.ppk</code>
-srcarg3	Plugin argument, in this case it specifies the sudo location: <code>/usr/bin/sudo</code>
-backupuser	Username with database backup/restore privileges, when the migration starts, ZDM will request for the password for this user, please provide the authentication token for the object storage. In this case the user is <code>bkpuser</code>
-rsp	Response file location, in this case the location is: <code>/u01/response/zdm_template.rsp</code>
-tgtauth	Authentication plug-in required to access the target database server. This let's ZDM know that a number n of plugin arguments are to come, specifying the specific authentications to be followed.
-tgtarg1	Plugin argument, in this case it specifies the <code>opc</code> user

-tgtarg2	Plugin argument, in this case it specifies the location of the public keys: /home/zdmuser/.ssh/zdm-servicenode.ppk
-tgtarg3	Plugin argument, in this case it specifies the sudo location: /usr/bin/sudo
-ignore	Ignore checks (ALL WARNING PATCH_CHECK)
-eval	Perform a dry run or evaluation of the migration without actually migrating the database. This allows to be sure that all migration phases will run smoothly. If there is any corrections to be made prior to migration, ZDM will alert the user.

Upon execution of the command, and password prompt, a job id will be generated:

```
./zdmcli migrate database -sourcedb db1_phx -sourcenode sourcezdm -srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode targetzdm
-targethome /u01/app/oracle/product/18.01
-backupuser bkpuser -rsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/zdmuser/.ssh/zdm-spk -tgtarg3 sudo_location:/usr/bin/sudo
-ignore ALL -eval

Enter source database db1_phx1fn SYS password:
Enter user "xxxxx" password:
Operation "zdmcli migrate database" scheduled with the job ID "4".
```

The second password that is requested, as per the backup user, refers to the Authentication token for the object store.

You can query the command progress with the following command:

```
./zdmcli query job -jobid 4
```

```

[zdmuser@zdm-servicenode bin]$ ./zdmcli query job -jobid 4
zdm-servicenode.x.x.x.x: Audit ID: 27
Job ID: 4
User: zdmuser
Client: zdm-servicenode
Scheduled job command: "zdmcli migrate database -sourcedb db1_phx1fn -sourcenode sourcezdm -srcauth
zdmauth -srcarg1 user:opc -srcarg2 identity_file: /zdmuser/.ssh/zdm-servicenode.ppk -srcarg3
sudo_location:/usr/bin/sudo -targetnode targetzdm -targethome /u01/app/oracle/product/18.0.0user
xxxxx -rsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2
identity_file:/home/zdmuser/.ssh/zdm-serg3 sudo_location:/usr/bin/sudo -ignore ALL -eval"
Scheduled job execution start time: 2020-X. Equivalent local time: 2020-X
Current status: SUCCEEDED
Result file path: "/u01/app/oracle/zdmbase/chkbase/scheduled/job-4-2020-05.log"
Job execution start time: 2020-05-06 20:11:49
Job execution end time: 2020-05-06 20:15:15
Job execution elapsed time: 3 minutes 25 seconds
ZDM_GET_SRC_INFO ..... COMPLETED
ZDM_GET_TGT_INFO ..... COMPLETED
ZDM_SETUP_SRC ..... COMPLETED
ZDM_SETUP_TGT ..... COMPLETED
ZDM_GEN_RMAN_PASSWD ..... COMPLETED
ZDM_PREUSERACTIONS ..... COMPLETED
ZDM_PREUSERACTIONS_TGT .... COMPLETED
ZDM_VALIDATE_SRC ..... COMPLETED
ZDM_VALIDATE_TGT ..... COMPLETED

```

Performing a Database Migration

To perform a database migration process, the `zdmcli migrate database` command must be executed, now without the `-eval` flag. For example, this is how the command would look like:

```

./zdmcli migrate database -sourcedb db1_phx -sourcenode sourcezdm -srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode targetzdm
-targethome /u01/app/oracle/product/18.01
-backupuser bkpuser -rsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/zdmuser/.ssh/zdm-spk -tgtarg3 sudo_location:/usr/bin/sudo
-ignore ALL
Enter source database sourcedb db1_SYS password:
Enter user "xxxxxxx" password:
Operation "zdmcli migrate database" scheduled with the job ID "5".

```

The second password that is requested, as per the backup user, refers to the Authentication token for the object store.

Upon execution of the command, and password prompt, a job id will be generated, you can query the command progress with the following command:

```

./zdmcli query job -jobid 5

```

Upon completion of the migration, this is how the output would look like:

```
zdmuser@zdm-servicenode bin]$ ./zdmcli query job -jobid 5
zdm-servicenode.x.x.x.x: Audit ID: 52
Job ID: 5
User: zdmuser
Client: zdm-servicenode
Scheduled job command: "zdmcli migrate database -sourcedb db1_phx1fn -sourcenode sourcezdm -srcauth
zdmauth -srcarg1 user:opc -srcarg2
identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode
targetzdm -targethome /u01/app/oracle/product/18.0.0.0/dbhome_1 -
backupuser xxxxx -rsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 id
entity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk -tgtarg3 sudo_location:/usr/bin/sudo -ignore ALL"
Scheduled job execution start time: 2020-04-15T20:23:59Z. Equivalent local time: 2020-05-06 20:23:59
Current status: SUCCEEDED
Result file path: "/u01/app/oracle/zdmbase/chkbase/scheduled/job-5-2020-05.log"
Job execution start time: 2020-05-06 20:24:19
Job execution end time: 2020-05-06 20:45:15
Job execution elapsed time: 20 minutes 55 seconds
ZDM_GET_SRC_INFO ..... COMPLETED
ZDM_GET_TGT_INFO ..... COMPLETED
ZDM_SETUP_SRC ..... COMPLETED
ZDM_SETUP_TGT ..... COMPLETED
ZDM_GEN_RMAN_PASSWD ..... COMPLETED
ZDM_PREUSERACTIONS ..... COMPLETED
ZDM_PREUSERACTIONS_TGT ..... COMPLETED
ZDM_VALIDATE_SRC ..... COMPLETED
ZDM_VALIDATE_TGT ..... COMPLETED
ZDM_OBC_INST_SRC ..... COMPLETED
ZDM_OBC_INST_TGT ..... COMPLETED
ZDM_BACKUP_FULL_SRC ..... COMPLETED
ZDM_BACKUP_INCREMENTAL_SRC .... COMPLETED
ZDM_DISCOVER_SRC ..... COMPLETED
ZDM_COPYFILES ..... COMPLETED
ZDM_PREPARE_TGT ..... COMPLETED
ZDM_SETUP_TDE_TGT ..... COMPLETED
ZDM_CLONE_TGT ..... COMPLETED
ZDM_FINALIZE_TGT ..... COMPLETED
ZDM_CONFIGURE_DG_SRC ..... COMPLETED
ZDM_SWITCHOVER_SRC ..... COMPLETED
ZDM_SWITCHOVER_TGT ..... COMPLETED
ZDM_NONCDBTOPDB_PRECHECK ..... COMPLETED
ZDM_NONCDBTOPDB_CONVERSION .... COMPLETED
ZDM_POSTUSERACTIONS ..... COMPLETED
ZDM_POSTUSERACTIONS_TGT ..... COMPLETED
ZDM_CLEANUP_SRC ..... COMPLETED
ZDM_CLEANUP_TGT ..... COMPLETED
```

FREQUENTLY ASKED QUESTIONS & COMMON ISSUES

F.A.Q

QUESTION	ANSWER
1. What authentication method should be used for object storage authentication?	<p>An authentication token can be created from the OCI Console. You can Navigate to the main Menu > Identity > Users.</p> <p>Then, you must select the user you want to use for the object storage authentication, click on the Auth Token option and click on generate token.</p> <p>Please copy the token upon creation, since it will not be accessible afterwards and if you want to access it you will need to generate a new one.</p>
2. What method of migration should be used for SE (standard edition) databases migration.	ZDM allows for Standard Edition as source databases. The migration method recommended is PHYSICAL_OFFLINE. This method uses the object store for the backup files. Also, starting with ZDM 21c, customers can leverage a Logical Workflow which can also benefit SE migrations.
3. Can ZDM be used to migrate database to on-premises physical servers ?	Yes, ZDM allows on-premises migration, providing that the target platform is an Exadata Database Machine.
4. Can we use ZDM to migrate databases that are hosted on different Linux versions than that target OS version ?	Yes, ZDM allows for cross OS version (Linux only) migration.
5. Can source and target database be on different versions?	<p>Oracle ZDM physical migration allows only for migration between databases of the same versions. The target database can be at a higher patch level, but then datapatch must be run post migration on the target database.</p> <p>Oracle ZDM logical migration does allow for cross version migration, hence providing a path for in-flight upgrades.</p>
6. Does ZDM perform non-cdb to cdb conversion while migration to the Oracle Cloud ?	Yes, Oracle ZDM does allow for this type of migration. The response file parameter NONCDBTOPDB_CONVERSION is required.
7. Does ZDM support cross platform migration ?	Yes, Oracle ZDM allows for Cross Platform Migration for databases hosted on AIX and Solaris. The supported method is LOGICAL_OFFLINE
8. Can ZDM used to migrate databases hosted in one OCI region to another OCI region ?	Yes, Oracle ZDM allows for cross OCI region migration.
9. Can ZDM used to migrate from OCI-C Database to OCI ?	Yes, one of Oracle ZDM's supported sources is databases hosted in OCI-Classic.
10. Will ZDM migrate Apex application reports?	No, ZDM does not perform this task.

<p>11. Can ZDM be instructed to not perform the swtichover, basically just setting up DG?</p>	<p>Yes, Oracle ZDM allows to pause the migration process on any given stage, and hence, the migration process can be paused before the role swap and swtichover phase. Upon executing the <code>zdm migrate database</code> command, the <code>-pauseafter</code> flag must be entered with the desired stage for pausing, in this case, <code>ZDM_CONFIGURE_DG_SRC</code></p>
<p>12. What are the supported OS versions for ZDM binaries?</p>	<p>Oracle ZDM runs on Oracle Linux 7</p>
<p>13. Do you provide interium patches for the ZDM , in-case if we are hitting a bug?</p>	<p>Oracle ZDM will release patches, addressing issues when required, if you are having an issue please file a service request and contact product management.</p> <p>In order to expedite resolution time, please follow the instructions in My Oracle Support Document: SRDC – Data Collection for Database Migration Using Zero Downtime Migration (ZDM) (DOC ID 2595205.1)</p> <p>https://support.oracle.com/epmos/faces/DocContentDisplay?id=2595205.1</p>
<p>14. Can ZDM be installed on the source or target server?</p>	<p>Yes, Oracle ZDM can be installed on any location, however, as a best practices is better to have ZDM run on a separate node. Further mode, Oracle GI must not be present on the node where the Oracle ZDM service node is to be deployed.</p>
<p>15. What are the OS packages required for ZDM install?</p>	<p>The following packages are required:</p> <ul style="list-style-type: none"> ○ <code>glibc-devel</code> ○ <code>expect</code> ○ <code>unzip</code> ○ <code>libaio</code> ○ <code>oraclelinux-developer-release-el7</code>
<p>16. Is it required to have the same databases patches between source and target databases?</p>	<p>This is recommended as a best practice, however, the target database can be at a higher patch level, but then datapatch must be run post migration on the target database.</p>
<p>17. What is the supported key format for ZDM to authenticate with source and target?</p>	<p>The supported format is RSA</p>
<p>18. What is storage types are supported according to the target cloud platform?</p>	<ul style="list-style-type: none"> • Oracle Database Cloud Service Bare Metal <ol style="list-style-type: none"> 1. OCI Object Storage • Oracle Database Cloud Service Virtual Machine <ol style="list-style-type: none"> 1. OCI Object Storage • Exadata Cloud Service <ol style="list-style-type: none"> 1. Oracle Zero Data Loss Recovery Appliance 2. NFS Storage • Exadata Cloud at Customer <ol style="list-style-type: none"> 1. Oracle Zero Data Loss Recovery Appliance 2. NFS Storage • Exadata On Premises

	<ol style="list-style-type: none"> 1. Oracle Zero Data Loss Recovery Appliance 2. NFS Storage <ul style="list-style-type: none"> • Autonomous Database (Logical workflow only) <ol style="list-style-type: none"> 1. OCI Object Storage
19. Do we need a license for ZDM software?	Oracle ZDM is free of charge and follows the following licensing agreement: https://www.oracle.com/downloads/licenses/sqldev-license.html#licenseContent
20. What are the supported target type's for migration?	<ul style="list-style-type: none"> • Oracle Database Cloud Service Bare Metal • Oracle Database Cloud Service Virtual Machine • Exadata Cloud Service • Exadata Cloud at Customer • Exadata Database On-Premises (Physical workflow only) • Autonomous Database (Logical workflow only)

Known Issues

All common issues are documented and updated periodically in Oracle Zero Downtime Migration's documentation, specifically on the product release note, Known Issues section:

<https://docs.oracle.com/en/database/oracle/zero-downtime-migration/>

Troubleshooting & Other Resources

For Oracle ZDM log review:

- ZDM Server Logs:
 - o Check - \$ZDM_BASE/crsdata/<zdm_service_node>/rhp/rhpserver.log.0
- Check source node logs
 - o - <oracle_base>/zdm/zdm_<src_db_name>_<job_id>/zdm/log
- Check target node logs
 - o - <oracle_base>/zdm/zdm_<tgt_db_name>_<job_id>/zdm/log

For all Oracle Support Service Requests related to Zero Downtime Migration, please be sure to follow the instructions in My Oracle Support Document:

- SRDC – Data Collection for Database Migration Using Zero Downtime Migration (ZDM) (DOC ID 2595205.1)
- <https://support.oracle.com/epmos/faces/DocContentDisplay?id=2595205.1>

Please review our Maximum Availability Architecture document for best practices related to Zero Downtime Migration.

- MAA Practices for Cloud Migration Using ZDM (Doc ID 2562063.1)
- <https://support.oracle.com/epmos/faces/DocContentDisplay?id=2562063.1>

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Oracle Zero Downtime Migration – ZDM Physical Migration Step by Step Guide
October, 2021

Author: Ameet Kumar Nihalani, Ricardo Gonzalez

Contributing Authors: Oracle RACPack Team, Oracle ZDM Development Team

